



Welcome United States Patent and Trademark Office

[Search Results](#)
[BROWSE](#)
[SEARCH](#)
[IEEE XPLORE GUIDE](#)

Results for "((firewall<in>metadata) <and> (remote<in>metadata))"

☒ e-mail

Your search matched 22 of 1235066 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

» Search Options

[View Session History](#)
[New Search](#)

Modify Search

>>

☐ Check to search only within this results set
Display Format: ☒ Citation ☐ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEEE JNL IEEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEEE CNF IEEE Conference Proceeding

IEEE STD IEEE Standard

Select Article information

**1. A new remote configurable firewall system for home-use gateways**

Mizuno, S.; Matsuura, K.; Yamada, K.; Takahashi, K.;
Consumer Communications and Networking Conference, 2005. CCNC. 2005 3-6 Jan. 2005 Page(s):599 - 601
Digital Object Identifier 10.1109/CCNC.2005.1405247

[AbstractPlus](#) | Full Text: [PDF](#)(1084 KB) IEEE CNF
**2. Secure remote access from office to home**

Kara, A.;
Communications Magazine, IEEE
Volume 39, Issue 10, Oct. 2001 Page(s):68 - 72
Digital Object Identifier 10.1109/35.956115

[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(664 KB) IEEE JNL
**3. Resource manager for Globus-based wide-area cluster computing**

Tanaka, Y.; Sato, M.; Hirano, M.; Nakada, H.; Sekiguchi, S.;
Cluster Computing, 1999. Proceedings. 1st IEEE Computer Society International 2-3 Dec. 1999 Page(s):237 - 244
Digital Object Identifier 10.1109/IWCC.1999.810830

[AbstractPlus](#) | Full Text: [PDF](#)(176 KB) IEEE CNF
**4. Remote physical device fingerprinting**

Kohno, T.; Broido, A.; Claffy, K.C.;
Dependable and Secure Computing, IEEE Transactions on
Volume 2, Issue 2, April-June 2005 Page(s):93 - 108
Digital Object Identifier 10.1109/TDSC.2005.26

[AbstractPlus](#) | Full Text: [PDF](#)(1048 KB) IEEE JNL
**5. Secure remote access to an Internet Web server**

Gilmore, C.; Kormann, D.; Rubin, A.D.;
Network, IEEE
Volume 13, Issue 6, Nov.-Dec. 1999 Page(s):31 - 37
Digital Object Identifier 10.1109/65.806985

[AbstractPlus](#) | Full Text: [PDF](#)(624 KB) IEEE JNL
**6. The NetScript active network system**

da Silva, S.; Yemini, Y.; Florissi, D.;

Selected Areas in Communications, IEEE Journal on
Volume 19, Issue 3, March 2001 Page(s):538 - 551
Digital Object Identifier 10.1109/49.917713

[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(200 KB) IEEE JNL

- ☐ **7. Remote physical device fingerprinting**
Kohno, T.; Broido, A.; Claffy, K.;
Security and Privacy, 2005 IEEE Symposium on
8-11 May 2005 Page(s):211 - 225
Digital Object Identifier 10.1109/SP.2005.18
[AbstractPlus](#) | Full Text: [PDF](#)(10200 KB) IEEE CNF

- ☐ **8. Port randomized VPN by mobile codes**
Shiraishi, Y.; Fukuta, Y.; Morii, M.;
Consumer Communications and Networking Conference, 2004. CCNC 2004. F
5-8 Jan. 2004 Page(s):671 - 673
[AbstractPlus](#) | Full Text: [PDF](#)(1281 KB) IEEE CNF

- ☐ **9. Eyeball Video Communications Platform**
Vass, J.; Khan, S.;
Multimedia Signal Processing, 2002 IEEE Workshop on
9-11 Dec. 2002 Page(s):396 - 399
[AbstractPlus](#) | Full Text: [PDF](#)(472 KB) IEEE CNF

- ☐ **10. Security mechanisms in high-speed networks**
Singhal, M.;
Computer Communications and Networks, 2000. Proceedings. Ninth Internati
on
16-18 Oct. 2000 Page(s):482
Digital Object Identifier 10.1109/ICCCN.2000.885533
[AbstractPlus](#) | Full Text: [PDF](#)(712 KB) IEEE CNF

- ☐ **11. Multiple intelligent agent supported Internet security system: Issues, cur
and a proposed approach**
Lin Zeng; huiqing Wang; Lee, M.K.O.;
Intelligent Processing Systems, 1997. ICIPS '97. 1997 IEEE International Conf
Volume 1, 28-31 Oct. 1997 Page(s):920 - 922 vol.1
Digital Object Identifier 10.1109/ICIPS.1997.672965
[AbstractPlus](#) | Full Text: [PDF](#)(184 KB) IEEE CNF

- ☐ **12. Secure mobile code computing in distributed remote environment**
Negm, K.E.A.; Adi, W.;
Networking, Sensing and Control, 2004 IEEE International Conference on
Volume 1, 21-23 March 2004 Page(s):270 - 275 Vol.1
Digital Object Identifier 10.1109/ICNSC.2004.1297447
[AbstractPlus](#) | Full Text: [PDF](#)(1503 KB) IEEE CNF

- ☐ **13. Collaborative visualization for supporting joint researches in ITBL projec**
Suzuki, Y.; Matsumoto, N.; Sai, K.;
Creating, Connecting and Collaborating through Computing, 2004. Proceeding
International Conference on
29-30 Jan. 2004 Page(s):187
Digital Object Identifier 10.1109/C5.2004.1314393
[AbstractPlus](#) | Full Text: [PDF](#)(188 KB) IEEE CNF

- ☐ **14. Secure remote control of field-programmable network devices**
Song, H.; Lu, J.; Lockwood, J.; Moscola, J.;
Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th A

Symposium on
20-23 April 2004 Page(s):334 - 335
Digital Object Identifier 10.1109/FCCM.2004.51
[AbstractPlus](#) | Full Text: [PDF](#)(74 KB) IEEE CNF

- ☐ **15. A system for fault detection and reconfiguration of hardware based activ.**
Bartzoudis, N.G.; Fragkiadakis, A.G.; Parish, D.J.; Nunez, J.L.;
On-Line Testing Symposium, 2004. IOLTS 2004. Proceedings. 10th IEEE Inter
12-14 July 2004 Page(s):207 - 213
[AbstractPlus](#) | Full Text: [PDF](#)(423 KB) IEEE CNF

- ☐ **16. A fault-tolerant approach to network security**
Lundell, J.;
Network Computing and Applications, 2001. NCA 2001. IEEE International Sy
8-10 Oct. 2001 Page(s):227
Digital Object Identifier 10.1109/NCA.2001.962536
[AbstractPlus](#) | Full Text: [PDF](#)(196 KB) IEEE CNF

- ☐ **17. FATIMA: a firewall-aware transparent Internet mobility architecture**
Mink, S.; Pahlke, F.; Schafer, G.; Schiller, J.;
Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE S
3-6 July 2000 Page(s):172 - 179
Digital Object Identifier 10.1109/ISCC.2000.860635
[AbstractPlus](#) | Full Text: [PDF](#)(692 KB) IEEE CNF

- ☐ **18. Personal Electronic Notebook with Sharing**
Hong, J.; Toye, G.; Leifer, L.J.;
Enabling Technologies: Infrastructure for Collaborative Enterprises, 1995., Pro
Fourth Workshop on
20-22 April 1995 Page(s):88 - 94
Digital Object Identifier 10.1109/ENABL.1995.484552
[AbstractPlus](#) | Full Text: [PDF](#)(732 KB) IEEE CNF

- ☐ **19. Real-time Intranet-controlled virtual instrument multiple-circuit power mc**
Chung-Ping Young; Wei-Lun Juang; Devaney, M.J.;
Instrumentation and Measurement, IEEE Transactions on
Volume 49, Issue 3, June 2000 Page(s):579 - 584
Digital Object Identifier 10.1109/19.850398
[AbstractPlus](#) | [References](#) | Full Text: [PDF](#)(436 KB) IEEE JNL

- ☐ **20. Internet-based distributed data acquisition system for fiber-optic sensors**
Bock, P.J.; Majumdar, S.; Bock, W.J.;
Instrumentation and Measurement Technology Conference, 2004. IMTC 04. P
21st IEEE
Volume 2, 18-20 May 2004 Page(s):885 - 890 Vol.2
Digital Object Identifier 10.1109/IMTC.2004.1351205
[AbstractPlus](#) | Full Text: [PDF](#)(481 KB) IEEE CNF

- ☐ **21. Telecommunication services for utility owners**
Cooley, K.;
Power Engineering Society Summer Meeting, 2000. IEEE
Volume 1, 16-20 July 2000 Page(s):583 - 584 vol. 1
Digital Object Identifier 10.1109/PESS.2000.867650
[AbstractPlus](#) | Full Text: [PDF](#)(96 KB) IEEE CNF

- ☐ **22. Emerging needs for continuous flow FOUP transport**
Brain, M.; Gould, R.; Kaempf, U.; Wehrung, B.;
Electronics Manufacturing Technology Symposium, 1999. Twenty-Fourth IEEE

18-19 Oct. 1999 Page(s):76 - 82

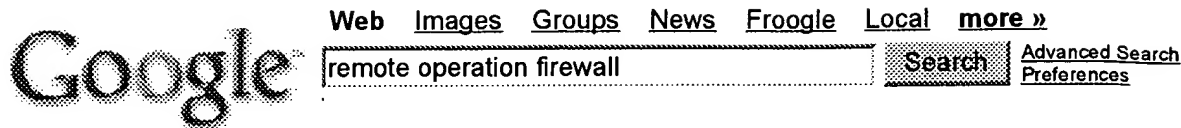
Digital Object Identifier 10.1109/IEMT.1999.804799

[AbstractPlus](#) | Full Text: [PDF\(404 KB\)](#) IEEE CNF



[Help](#) [Contact Us](#) [Privacy & :](#)

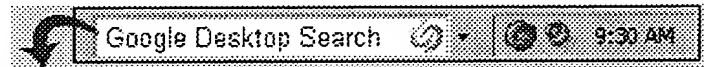
© Copyright 2005 IEEE -

**Web**Results 1 - 10 of about **1,870,000** for **remote operation firewall** . (0.40 seconds)**RealVNC - Frequently Asked Questions**VNC uses an open platform-independent protocol to allow desktop **remote** control.... Many organisations **operate** **firewalls** to reduce the risk of intrusion by ...www.realvnc.com/faq.html - 22k - Sep 17, 2005 - [Cached](#) - [Similar pages](#)**Firewalls/VPN: Juniper (Formerly NetScreen)**Advances in **remote operations** include a rescue button that any on-site personnel... NetScreen-Global PRO provides central **firewall** and VPN configuration ...www.conqwest.com/solutions_netscreen.asp - 53k - [Cached](#) - [Similar pages](#)**How to troubleshoot WMI-related issues in Windows XP SP2**When you perform a **remote** WMI **operation** from a computer A to computer B, ...Right-click Windows **Firewall**: Allow **remote** administration exception, ...go.microsoft.com/fwlink/?LinkID=44305 - [Similar pages](#)**Juniper Networks :: Products & Services**Juniper's security products include integrated **firewall**/IPSec VPN devices ...while also easing **remote** management through advanced **operations** tools. ...www.juniper.net/products/ - 22k - Sep 17, 2005 - [Cached](#) - [Similar pages](#)**Remote Operations via the Internet**K6WR/4 **operating** through the W4MQ **remote** in Reston, VA ... Please disable any **firewall** while you are testing the sound setup. Then when the sound works, ...www.w4mq.com/help/faq.html - 19k - [Cached](#) - [Similar pages](#)**[PDF] Systemwalker Live Help Systemwalker Live Help**File Format: PDF/Adobe Acrobat - [View as HTML](#)Provide system support via the Internet through **firewalls** ... **Remote Operation**.- **Operate** the user's PC with keyboard and mouse ...www.fujitsu.com/downloads/SYSWLK/pdf/live_i.pdf - [Similar pages](#)**Remote Desktop Tips for the SOHO User**You can verify correct **operation** of **Remote Desktop** by connecting from another PC... The Windows XP SP2 Windows **Firewall** can be configured to allow **Remote** ...theillustratednetwork.mvps.org/RemoteDesktop/RemoteDesktopSetupandTroubleshooting.html - 25k -[Cached](#) - [Similar pages](#)**NOVELL: Cool Solutions: AppNote: ZENworks with NAT and ...****Remote** Management and Inventory work well in a **firewall** environment, ... In thiscase the **Remote** Management **operations** works only if you have a static NAT, ...www.novell.com/coololutions/appnote/8016.html - 25k - [Cached](#) - [Similar pages](#)**ROP - Raster OPERATION, Remote OPERATION, Risc OPERATION**AUDITMYPC MAIN · **FIREWALL** TEST · SPYWARE REMOVAL · POPUP TEST · SOFTWARE AUDIT ·INTERNET SPEED TEST ... 2) ROP is also derived from **Remote OPERATION** ...www.auditmypc.com/acronym/ROP.asp - 25k - [Cached](#) - [Similar pages](#)**NETGEAR - ProSafe™ VPN Firewall 8 w/8 Port 10/100 Switch**

Remote administration capability makes connecting to multiple sites a breeze.
All NETGEAR ProSafe Firewalls are VPNC tested and certified to ensure ...
www.netgear.com/products/details.php?ID=FVS318 - 33k - [Cached](#) - [Similar pages](#)

Google

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)



Free! Instantly find your email, files, media and web history. [Download now.](#)

remote operation firewall [Search](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [Hurricane Katrina Resources](#) - [About Google](#)

©2005 Google

Session 25

Mobile Communication Networks VI

Trusted and Active Protocol over a Distributed Architecture in ATM Networks with Software Agents¹

José Luis González-Sánchez (*) and Jordi Domingo-Pascual (**)

(*) University of Extremadura. Escuela Politécnica de Cáceres Avda. Universidad S/N. (10.071) Cáceres (Spain)

Phone: +34 927 257 259 Fax: +34 927 257 202 E-Mail: jlgs@unex.es

(**) Polytechnic University of Catalunya. Campus Nord, Modul D6, Jordi Girona 1-3 (08034) Barcelona (Spain)

Phone: +34 93 401 6981 Fax: +34 93 401 7055 E-Mail: jordi.domingo@ac.upc.es

Abstract-TAP (Trusted and Active PDU transfers) is a new distributed architecture and protocol for ATM networks that provides assured transfers to a set of privileged VPI/VCI. The distributed architecture proposes a new Extended AAL type 5 (EAAL-5), manages the privileged connections and offers an improvement in the performance when network connections cause some cell loss by taking advantage of the idle time in the traffic sources to carry out the retransmissions of CPCS-PDU-EAAL5. The trusted protocol is supported by our AcTMs (active ATM switch) model that we have equipped with hardware techniques and active software to achieve our objectives. Several simulations (using ON/OFF sources) demonstrate the effectiveness of the mechanism that recovers the congested PDUs locally at the congested switches with better goodput in the network. Also, the senders are alleviated of negative end-to-end retransmissions. The TAP is an active and distributed architecture in the sense that our protocol implements several active coordinated and self-collaborative software agents.

1. INTRODUCTION AND RELATED WORK

Reliability in ATM networks is provided by the Header Error Control (HEC) field of 8 bits in the header of the ATM cells and by the Cyclic Redundancy Check (CRC) in the Common Sublayer-Protocol Data Unit (CS-PDU). Error control is performed end-to-end by the terminals. The main problem is that a single cell loss causes a reassembly CRC error at AAL-5 level, which in turn leads to a retransmission of a complete PDU (i.e., IP datagram).

ATM networks experience three types of errors [1-2]: cell losses due to congestion in switches; corruption of data portions due to bit errors, and switching errors due to undetected corruption of the cell header. We note that congestion is by far the most common type of error, and here is where we want to improve the trusted transfers with TAP.

Current literature describes three basic techniques to achieve reliability: ARQ [3] (Automatic Repeat Request), FEC [4-6] (Forward Error Correction) and hybrid mechanisms of ARQ in combination with FEC.

While ARQ adds latency (due to the cost of NACK) and implosion, FEC adds overhead and thus the redundant code added by this method is useless when the network is experiencing congestion. Hence, ARQ may not be suitable for applications with requirements of low latency, and FEC performs worse in networks with low bandwidth or which experience frequent congestion. In our architecture we

adopted ARQ with NACK (using RM cells) to alleviate the effect of implosion. Support for reliable multipoint cannot be based on retransmissions from the source. In TAP, the intermediate active nodes carry out the retransmissions.

The most commonly proposed congestion control schemes to improve throughput and fairness, while minimizing delay in ATM networks, are the Random Cell Discard (RCD), Partial Packet Discard (PPD), Early Packet Discard (EPD) [7], Early Selective Packet Discard (ESPD), Fair Buffer Allocation (FBA) and Random Early Detection (RED). TAP uses a modified version of EPD (which we have denominated Early Packet Discard and Relay, EPDR) to alleviate the effect of congestion and packet fragmentation.

Nowadays, congestion control is achieved by delegated relay on end-to-end protocols, such as TCP. This is an easily implemented technique at high speeds and also simplifies the switches, but all the network is overcharged with retransmissions and does not achieve protection against egoistic traffic sources. Fair bandwidth schemes protect wellbehaved sources from misbehaved customer ones, and allow a diverse set of end-to-end congestion control mechanisms. We provide support for fair bandwidth allocations based on a delegated and modified WFQ (Weighted Fair Queueing) [8] scheme to reduce its implementation complexity.

ATM Adaptation Layer type 5 (AAL-5) has been developed to support transfers of non-assured data user frames, where the lost and corrupted Common Part Convergence Sublayer Service Data Unit (CPCS-SDU) cannot be solved with retransmission [1]. We propose EAAL-5 as an extended and enhanced native AAL-5. EAAL-5 is part of TAP that supports assured service with retransmissions and is also compatible with native AAL-5. In this paper we propose a mechanism to take advantage of the idle periods in the data sources to retransmit the Common Part Convergence Sublayer PDU (CPCS-PDU) of EAAL-5.

Active, open and programmable networks is a new technical area [9-19] to explore ways in which network elements may be dynamically reprogrammed by network managers, network operators or general users to accomplish the required QoS and other features such as customized services. This offers attractive advantages, but also important challenges in aspects such as performance, security or reliability. Hence, this is an open issue for research and development in customized routing and protocols, whether to

¹ This work is sponsored in part by the CICYT under Grant No. TEL99-1117-C03-03 and by the Regional Government of Extremadura (Education, Science and Technology Council).

move the service code (placed outside the transport network) to the network's switching nodes. Many of the advantages of active protocols are achieved by installing active nodes at strategic points. Concepts such as active networks, protocol boosters or software agents are proposed and developed for IP networks; however the proposals are insufficient for ATM networks.

We have simulated and studied transfers combined with the active switch and other non-active ATM switches to constitute a VPN (Virtual Private Network) as we can see in Fig. 1. Our goal is to use the TAP architecture to approximate the functionality of a network in which all switches do not need to implement TAP. Java Development Kit V1.2.1 has been the language and environment used to implement TAP due to the special characteristics offered by Java.

Section 2 describes TAP architecture. In sections 3 and 4, we present AcTMs, our prototype of an active switch, that support the architecture and the protocol. Section 5 describes and outlines our work currently in progress to enhance TAP. Finally we offer some concluding remarks in section 6.

II. GENERAL DESCRIPTION OF TAP ARCHITECTURE

AAL-5 was proposed [1] to reduce overhead introduced by AAL3/4. The CPCS-PDU format of native AAL-5 and EAAL-5 has equal fields. The tail of PDU has 4 fields. The CPCS-UU (User-to-User indication) field is used for the transfer of CPCS user to user information. The CPI (Common Part Indication) octet is used to align 64 bits to the CPCS-PDU tail. TAP utilizes these two octets as the PDU sequence number, which is assigned end-to-end by the EAAL-5 user. The CRC is used as in AAL-5 to detect bit errors in the CPCS-PDU. The value of CRC is calculated including all the fields of the CPCS-PDU. The sequence number in PDUs is preserved end-to-end to avoid recalculating the CRC and modifying the tail of the CPCS-PDUs.

To implement NACK we use the standard Resource Management (RM) cells, without fixed frequency but generated when a switch is congested. This is to alleviate the negative overhead effect due to a fixed number of RM cells that will waste bandwidth.

When congestion is detected, EPDR discards a PDU. Then the CCA Agent searches for the discarded PDU in the DMTE. If this PDU is not in the local DMTE, the RM Agent of the active node generates a RM cell which is transmitted backwards to the upstream active switch indicating the sequence number of the discarded PDU.

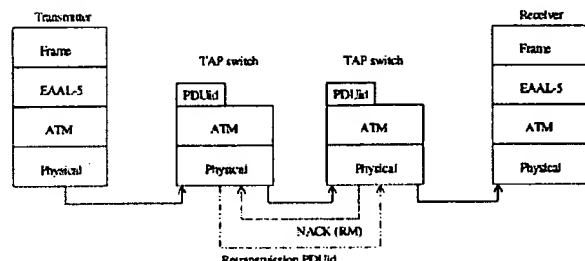


Fig. 1. Virtual Private Network with TAP.

The RM must also contain the VPI/VCI to identify the connection that has experienced discard problems. This mechanism is required to relate the sources of traffic with their I/O ports to alleviate the effect of equal values in VPI/VCI at different ports. Octets 22 to 51 of the RM cell [14] store the identifier Port/VPI/VCI/PDUid of requested PDUs.

When the RM cell arrives at an active switch, the TAP searches the requested PDU and, if it is still in DMTE, then the PDU is retransmitted, as long as the idle time for the connection is sufficient.

When a NACK (RM cell) arrives at a non-active switch this only processes the RM cell and resends it to its neighboring switch in the direction of the closest upstream active switch. The non-active switches do not have DMTE to retrieve PDUs and their function is only to send (or resend) PDUs forward to their destination and also send NACKs backwards to the active nodes.

To conclude this point we emphasize that TAP cannot offer complete reliability, but assures and recovers an important number of PDUs that otherwise would be lost by congestion. The mechanism also guarantees that there are no end-to-end retransmissions but between active peer nodes. The retransmission mechanism generates unordered PDUs at the receiver. The protocol offers two kind of service. The first one named SEQ (sequential) sorts the PDUs and when it detects a sequence failure, assumes that the PDU is lost and leaves the retrieval to protocols of upper layers (i.e. TCP). The second type of service is unordered (connectionless) and does not do any kind of sorting. These two services are offered by the proposed EAAL-5.

III. AcTMs (Active ATM Switches) NODES

Fig. 2 plots the TAP architecture including the DMTE memory and the agents. The trusted sources generate their flow that arrives at the DMTE and which is processed to the output buffer that multiplexes the cells to the next switch.

The architecture of the ATM switch is similar to an output buffered switch with VC-merging capabilities. We propose the AcTMs model of switch able to support the TAP architecture and protocol, which has the next main sections.

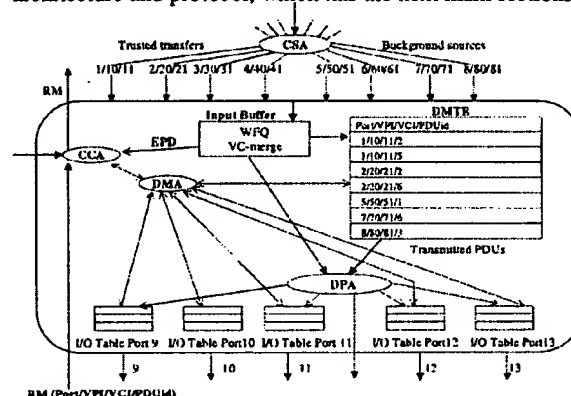


Fig. 2. TAP architecture over AcTMs switch.

A. DMTE (Dynamic Memory of Trusted EAAL-5 PDUs)

The DMTE is the key module. It behaves as a common shared memory and VC-merge buffer at the same time. The main function of this memory is to store temporarily the PDUs in the active switch after they have been transmitted to the output buffer, so that they can be requested in retransmissions. The TAP protocol keeps a copy of each PDU that arrives at the active switch (while VC merging is performed). Several PDUs are stored for each privileged connection. When a complete PDU arrives at the DMTE it is "copied" completely into the output buffer. The PDU remains in the DMTE until free space is needed for storing a new PDU.

Due to the big size of PDU-AAL-5 (up to 65,535 bytes), and the potential high number of connections (VPI/VCI), the size of the DMTE may be excessive. This is the reason why we limit the number of stored PDUs for each connection, and furthermore, only support a reduced number of privileged VCIs with trusted transfers. The traffic of a connection is more trusted when the DMTE stores more PDUs of this connection. But the size of DMTE also depends on the size of the PDUs, and we know that this is variable. We have calculated the required size of the memory to guarantee a concrete number of connections. Table I shows some of the obtained results and we can see that in order to offer trusted transfers to 1000 sources it only needs 3 Mbytes of memory that stores 2 PDUs of 1500 bytes for each privileged connection.

The TAP accesses the DMTE through an index consisting of the port number, the PDUid (which corresponds to the UU and CPI fields in AAL5) and the VPI/VCI, and we have implemented different mechanisms to optimize the management and storage of PDUs.

While a PDU from the DMTE is being retransmitted, if there is an incoming PDU with the same VPI/VCI and no free space is available in the DMTE, the incoming PDU is discarded. It is similar to a loss caused by the lack of VC-merge buffers.

B. Input buffer

As we can see in Fig. 2, the ATM cells arrive and are multiplexed over the input buffer where the cells are reassembled to build the PDUs. We use a size of 3000 octets, but the TAP simulator provides features to customize the number of cells that the input buffer stores.

TABLE I
REQUIRED DMTE SIZE TO GUARANTEE X CONNECTIONS

PDUs stored per connection	PDU size	Trusted connections	DMTE size
3	64 Kb	10	1.9 Mbytes
3	64 Kb	20	3.8 Mbytes
3	1.5 Kb	10	45 Kbytes
3	1.5 Kb	100	450 Kbytes
3	1.5 Kb	1000	4.5 Mbytes
2	1.5 Kb	1000	3 Mbytes

Every traffic management scheme requires queue management. Different methods for managing queues have different effects on traffic that flows through the queues. We know that *Work conserving systems* (FIFO) send the PDUs once the switch has completed the time of service. Hence, the server may not be idle if there are PDUs in a queue. On the other hand, the *Non-work conserving schemes* waits a random amount of time before serving the next PDU in a queue, even if PDUs are waiting in the queue. While packet switching networks use *window-based flow control* (FIFO), the high speed networks need *rate-based mechanisms* and use work-conserving services and mechanisms such as Fair Queueing (FQ). FQ waits n-1 bits times before sending and has the problem that every source has the same fraction of bandwidth. However Weighted Fair Queueing (WFQ) offers strong performance guarantees. Although algorithms designed to achieve fair bandwidth allocations provide many desirable advantages for congestion control, their implementation complexity (per-flow scheduling, per-flow buffer management and per-PDUs classification) is an important obstacle to their application in high-speed networks. We propose a WFQ scheme that works by delegation over the CSA agent as we can see in section IV.

C. Input/Output Tables

Each output port in AcTMs has its corresponding Input/Output table that stores the access index to the DMTE. When a PDU is sent to its output port, the I/O table that stores the *InPort/VPIIn/VCIIn/PDUid/VPIOut/VCIOut/OutPort* index is previously updated. This avoids equal PDUidentifiers and also provides direct access to PDUs into DMTE.

IV. MULTI-AGENT AND DISTRIBUTED SYSTEM

We bring active characteristics to TAP through hardware mechanisms and software techniques. TAP architecture includes the CCA agent (Control Congestion Agent) to manage the retransmission requests between peer active switches, the DMA agent (Dynamic Memory Agent) to recover PDU from the DMTE and also the CSA and DPA agents.

An active network is a programmable network that allows code to be loaded dynamically into network nodes at run-time. The literature on active networks studies several mechanisms to obtain advantage from active nodes. However, the proposals are insufficient for ATM networks and references [10-19] are some examples of this recent research in ATM.

There is no consensus on deciding when a network is active. There are two great tendencies: *a network is active if it incorporates active nodes with the capacity to execute a user's program*, or *if it implements mechanisms of code propagation*. The TAP architecture is active in both trends, because it provides active nodes at strategic points that implement an active protocol to allow user's code to be loaded dynamically into network nodes at run-time. TAP also provides support for code propagation in the network thanks to the RM cells. TAP is also a distributed architecture in the sense that the protocol uses several active coordinated and self-collaborative agents.

The ATM switch of our model network is an output buffered switch that just reads VPI/VCI information from arriving cells and forwards them to the corresponding output port. But we equipped this switch with active hardware (as we just see in section III) and software techniques (as now we will see). The TAP architecture uses four agents to perform the following functions.

A. CSA (Class of Service Agent)

This programmable agent supports by delegation the WFQ scheme. The TAP protocol extends the flow-based WFQ functionality to provide support for user-defined classes of service. CSA allows the definition of classes of service with a list of parameters for each class such as PCR, size of PDUs, QoS, exact amount of bandwidth, VPI/VCI identifiers, Ton, Toff, queue limit size, etc. When the list of parameters is defined in the transmitter, the CSA is coordinated with the following agents to guarantee the class of service defined end-to-end and the WFQ scheme provide congestion control and allocates bandwidth in a fair manner.

B. CCA (Control Congestion Agent)

The CCA programmable agent controls congestion based in EPDR and other algorithms that the network manager can choose. This agent monitors the output buffer and, when the occupancy is above the threshold, it discards any new incoming PDUs (packets). The last EAAL-5 cell contains the VPI and VCI in the header and the PDUid in the trailer of the EAAL-5-CPCS-PDU. The complete PDU is discarded as in EPD but the information about the VPI, VCI and PDUid is used to generate a request for the retransmission of this PDU. If the requested PDU is still in the local DMTE, it may be recovered and resent to the output buffer. Otherwise, the requests must be forwarded to the upstream active switch.

Fig. 3 shows the EPD algorithm modified to carry out the retransmissions of PDUs when a congestion is detected at input buffer (Early Packet Discard and Relay).

```

When a cell arrives at an ATM buffer:
  if the cell's VPI/VCI belongs to drop-list
    if the cell is an EOM cell
      if Queue_Length < Buffer_Size
        insert the cell into buffer
        get PDUid
        generates RM cell (Request PDUid)
      else
        discard the cell
    else
      remove the VPI/VCI from the drop-list
  else
    discard the cell
else
  if Queue_Length < Threshold
    insert the cell into buffer
  else if (BOM cell or (the buffer is full))
    discard the cell
    capture the VPI/VCI into drop-list
  else
    accept the cell into the single buffer
  
```

Fig. 3. Early Packet Discard and Relay Algorithm.

Another important point is that CCAs do not perform a protocol in the classical way; that is, there is only one opportunity to recover a PDU. No sliding windows or timeout retransmissions are used in this proposal.

C. DMA (Dynamic Memory Agent)

The DMA agent is the access point to DMTE memory. The CCA is coordinated with the Dynamic Memory Agent to request retransmission of the PDU to the peer active switches. The function of the CCA agent is to generate native RM cells that are transmitted backwards to the upstream active switch. Non-active switches will recognize the RM cells as TAP RM cells and will not take any action on them; they will simply forward the RM cells.

When a CCA agent receives an RM cell it takes on the task of looking for the requested PDU in the DMTE memory using *Port/VPI/VCI/PDUid* as the index. But this work is delegated over the DMA agent that uses the I/O tables to search the access index to the DMTE memory. If the PDU is still there it means that the connection cell flow presents an idle period and the PDU may be recovered. The PDU is sent to the DPA agent that dispatches it to their correspondent output port. When the PDU is not yet at DMTE the DMA agent notifies the CCA agent that is responsible for generating a RM cell backward to the previous ATM switch.

We should recall that PDUids are assigned end-to-end for each VCC and there is not any change for misinterpretation of the requested PDU. If the cell flow of the connection is dense (very short idle periods between successive PDUs) then the new incoming PDUs will use the DMTE and the "old" PDUs will be removed. We are currently working on the use of RM cells as a transport mechanism to carry out code propagation between active nodes. This code contains instructions to optimize the retransmission of PDUs in multipoint connections. The CCA agents utilize these instructions to inspect the distribution tree at width providing better goodput in retransmissions.

D. DPA (Dispatcher PDUs Agent)

This agent takes the complete PDUs of the input buffer and, when their Input/Output buffer is updated, the PDU is sent to the correct output port. DPA guarantees that the PDUs are sent completed to the output port. The congestion of the output port and also, the negative effect of merge cells belonging to different PDUs or sources, are avoided.

V. PERFORMANCE EVALUATION

Previous work [18,19] has presented and demonstrated the good behavior of RAP and TAP protocols over TAP architecture. We have simulated several software techniques to introduce active characteristics in switches. These mechanisms control and manage the privileged VCI and we will also offer an active mechanism to retrieve PDUs querying neighboring active switches and to search optimized paths when a PDU is retransmitted.

The simulation allows us to define the congestion probability in transmitters, receivers and each ATM switch. When a node is undergoing congestion, it then requests the retransmission (NACK) of the corresponding PDU.

The simulation also permits the user to introduce variable values such as ON/OFF traffic source parameters, the number of transmitters and receivers, the number of non-active switches, size of DMTE and input buffer, etc. Also, the simulator offers the operator the management of the parameters of programmable agents such as CSA and CCA.

In our simulation to analyse ATM cell loss we have used ON/OFF (bursty) traffic sources. The ON/OFF model [20,21] is used to characterise ATM traffic per unidirectional connection. Fig. 4 shows this model as a source which either actively sends (ON state) CSCP-PDU-AAL-5 data for some time T_{on} at a traffic rate R or PCR (Peak Cell Rate) or is silent (OFF state) producing no cells for some time T_{off} .

The source also periodically generates empty time slots. We use in all examples a CSR (Cell Slot Rate) of $C=353,208$ cell/s since our network model uses 155.52 Mbit/s links. When the CAR is less than the CSR, there are empty slots during the active states as we can see in Fig. 4.

The cell inter-arrival time $1/CAR$ is the unit of time for the ON state, and the mean duration in the active state is

$$T_{on} = (1 / CAR) * X_{on} \quad (1)$$

Also, the mean duration in the silent or idle state is

$$T_{off} = (1 / CSR) * X_{off} \quad (2)$$

Empirical studies [20] demonstrate that $T_{on} = 0.96$ seconds, and $T_{off} = 1.69$ seconds, and we use these values in the simulation, although we have used other values to analyze its effect over TAP. We use these and other formulae to implement the sources of simulations. Note that we have varied some of these parameters to analyse the behaviour of the TAP when it changes the scenario and the source traffic descriptors as we show in this section.

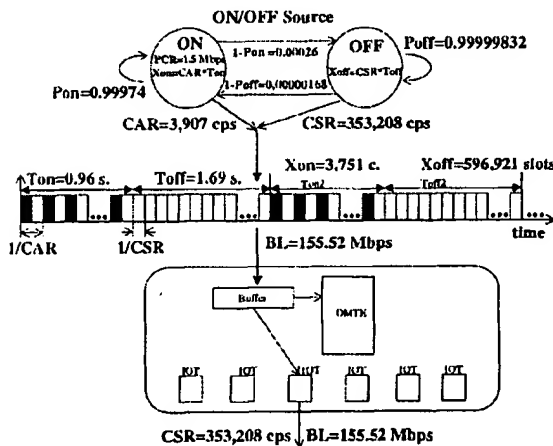


Fig. 4. Cell pattern for a single ON/OFF source and example of simulation.

Table II shows the maximum and minimum source traffic descriptors used in our simulation. We utilize a process that switches between an idle (silent) state, and the active state (sojourn time) which produces an average fixed rate of cells (between 64 Kbits/s to 25 Mbits/s) grouped in PDUs of 1,500 bytes. During the ON states this process generates cells at a cell arrival rate CAR.

Fig. 5 shows several identical sources, each operating independently over an AcTMs switch equipped with an ATM buffer of finite size X bytes and service capacity C cells/second receiving cells since the ON/OFF sources. The Peak Cell Rate is PCR cells/second, so the Mean Cell Rate (MCR) for each ON/OFF source is,

$$MCR = PCR (T_{on} / (T_{on} + T_{off})). \quad (3)$$

The probability that the source is active, or activity factor, is,

$$AF = MCR / PCR = T_{on} / (T_{on} + T_{off}). \quad (4)$$

We can calculate how many times the peak rate PCR fits into the service capacity C , denoted by S ,

$$S = C / PCR. \quad (5)$$

In this way, in the example shown in Fig. 4,

$S = 353,208 / 3,751 = 94.1$ ON/OFF trusted sources with $PCR = 1.5$ Mbps and multiplexed over a link bandwidth of 155.52 Mbps.

So, we will have enough aggregated T_{off} time to retrieve PDUs when we have not exceeded 94 multiplexed sources. This gives the maximum number of sources that we can have in the system to take advantage of the inactivity time to retransmit congested PDUs.

When TAP protocol detects that the aggregated PCR of sources exceed the service capacity C , with burst scale queuing, it does not request retransmissions of PDUs. So, this feature is accomplished by the CCA agent which, before requesting retransmissions, checks that these are possible in order to avoid waste network resources.

TABLE II
SOURCE TRAFFIC DESCRIPTORS ON/OFF

Source traffic descriptor	Parameter	Minimum	Maximum
Bandwidth Source	BS	64 kbit/s.	25 Mbit/s.
Cell arrival rate	CAR	167 cells/s.	65,105 c/s.
Cell inter-arrival time	1/CAR	6 ms.	15 μ s.
Bandwidth link	BL	155.52 Mbps.	622 Mbit/s.
Cell slot rate	CSR	353,208 cell/s.	1,412,648 cell/s.
Service time per cell	1/CSR	2.83 μ s.	0.70 μ s.
Active time period	T_{on}	0.96 s.	1 s.
Mean number of cells in an active state	X_{on}	160 cells	65,105 cells
Time in idle state	T_{off}	1.69 s.	2 s.
Mean number of empty slots in idle states	X_{off}	596,921 cell slots	2,825,296 cell slots

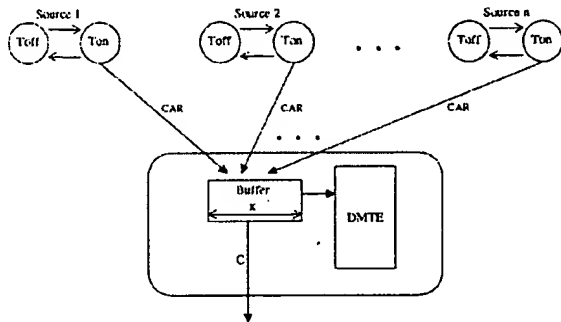


Fig. 5. Multiple ON/OFF sources.

We now report some results from the simulation of the TAP protocol.

Fig. 6 shows the effect of varying PCR between 86 and 2,667 cells per second (33,000 Kbps to 1 Mbps respectively). In this simulation we fixed the congestion probability at 10^{-3} . We use an input buffer of 3,000 octets and the DMTE stores 2 PDUs of 1,500 bytes for each connection.

The value for PCR is 64 Kbps (167 cells/s.); Ton=0.96 s.; and Toff=1.69 s., over the 50 total PDUs discarded by congestion, 50 PDUs are retrieved via TAP. Also, when PCR=56 Kbps and 33 Kbps, TAP retrieves all the congested PDUs. Thus, the performance is optimized (50 retrieved PDUs out of 50 congested PDUs) since all the lost PDUs are retrieved and there are no DMTE failures (all the requested PDUs are in the DMTE).

As we can see, when the arrival rate is low, the number of retrieved PDUs increases. When the PCR increases, 256 Kbps, TAP retrieves 48 out of 50 PDUs, but the 2 lost PDUs are not requested because the protocol detects insufficient idle time (Toff) to do the retransmission. We can see how the number of NACKs not sent (Not requested PDUs) is greater when the PCR value increases. In this way, the network is not over-charged with useless retransmissions when there is not sufficient aggregate Toff.

We note that at high PCR (1 Mbps) the number of retrieved PDUs is 47 and also the 3 not retrieved PDUs are not requested. As we can see the goodput is optimized when the number of trusted sources do not exceed the service capacity C (see Fig. 4 and 5).

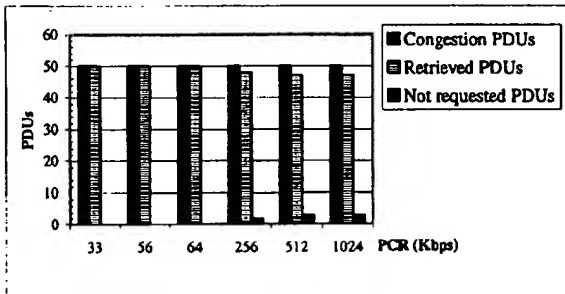


Fig. 6. Number of retrieved PDUs for different PCRs.

Fig. 7 shows the results of varying the idle time (Toff) between 0.1 and 2 seconds. We now use 10 ON/OFF sources over a bandwidth link of 25 Mbps. Each source generates 500 PDUs at PCR=1 Mbps. We fixed a Cell Loss Rate of 5 % over the total emitted PDUs and we have a constant value of 25 congested PDUs. As we can see, when the aggregated Toff is sufficient (0.5 seconds), all the congested PDUs are retrieved (25 retrieved over 25 congested). When the Toff is less than 0.5 s. the retrieved PDUs fall to 12 PDUs at 0.3 s., and 3 retrieved PDUs at 0.1 s. So we noted that, when there is insufficient Toff, the number of unretrieved PDUs increases, but TAP guarantee the goodput since the unrecoverable PDUs are not requested to avoid overcharging the network.

Another scenario consists of 1 source node, 1 active ATM switch, n non-active switches and 1 destination node. When a NACK arrives at a non-active switch, this also transfers the RM cell to the next switch. When the RM arrives at the active switch this uses the DMTE to retransmit the requested PDU. This scenario is the same as above, only the number of non-active switches varies. In this configuration we have simulated the protocol with several non-active switches and the results obtained show no changes. Only the delay in transmissions varies due to propagation times, but the index of retrieved PDUs is maintained as we have already shown.

Previous work [18,19] presents a point-multipoint configuration consisting of 1 source node, 1 active ATM switch, n non-active switches and n destination nodes. This is equal to the above basic scenario, only the number of destination nodes in multipoint connections varies. At present we are working to achieve multipoint connections to TAP. If we consider the above results we can see intuitively that the total delay will change. Also the amount of DMTE memory required increases in active switches to manage the VPI/VCI of n connections.

However, we shall now describe several aspects on which we are working to achieve better goodput.

We will consider other source traffic descriptors such as SCR (Sustainable Cell Rate) and MBS (Maximum Burst Size). With these parameters we can characterize the traffic better. Also, like most applications used in the TCP protocol for transmission data in frame based structures, we are working to implement the Guaranteed Frame Rate (GFR) [22] service class to provide a minimum service guarantee for

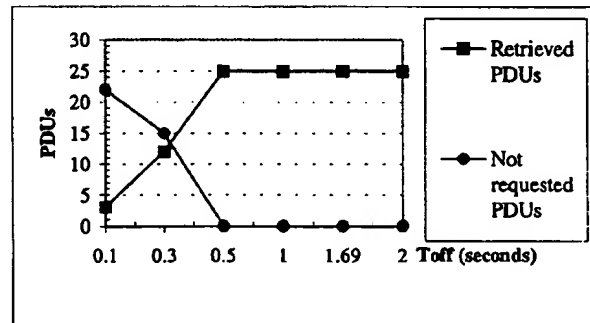


Fig. 7. Effect of variation Toff.

UBR, VBR and ABR services. In order to support GFR we will simulate sources with a Minimum Cell Rate (MCR) guarantee for a given MBS and Maximum Frame Size (MFS). With the GFR service class TAP will guarantee that is able to distinguish eligible and non-eligible frames and also to discard cells properly. We are currently working to enhance the architecture including other intelligent agents to characterize the traffic and their class of service.

VI. SUMMARY

In this paper we have presented TAP as the architecture for an active protocol that can take advantage of suitably equipped active ATM switches. TAP manages a set of privileged VCIs to improve trusted connections when the switches are congested. To achieve these active characteristics we use AcTMs (active ATM switches) equipped with little support hardware and software agents of reduced implementation complexity. We have verified that it is possible to retrieve an important number of PDUs only with DMTE and a reasonable additional complexity of the AcTMs switches supported by software agents that implement variants of EPD to solve congestions, and WFQ to achieve fair allocation. The retransmission mechanism is based on ARQ with NACK that generates RM cells to request PDUs. Our simulations demonstrate that the intuitive idea of taking advantage of silent states in ON/OFF sources is valid. Thus we can achieve better goodput and QoS in ATM networks.

REFERENCES

- [1] Recommendation I.363.5, "B-ISDN ATM Adaptation Layer Specification, Type 5," *ITU-T*, 08/1996.
- [2] Recommendation I.371.1, "Traffic Control and Congestion Control in B-ISDN," *ITU-T*, 06/1997.
- [3] Stephan Block, Ken Chen, Philippe Godlewski, and Ahmed Serhrouchni, "Design and Implementation of a Transport Layer Protocol for Reliable Multicast Communication," *Université Paris*, <http://www.enst.fr/~block/srmtp>, 1998.
- [4] Jörg Nonnenmacher, M. Lacher, M. Jung, E. Biersack, and G. Carle, "How bad is Reliable Multicast without Local Recovery?," *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE*, Vol. 3 pp. 972-979, 1998.
- [5] Ernst W. Biersack, "Performance Evaluation of Forward Error Correction in an ATM Environment," *IEEE Journal on Selected Areas in Comm.*, vol. 11, No. 4, pp. 631-640, May. 1993.
- [6] H. Esaki, G. Carle, T. Dwight, A. Guha, K. Tsunoda, and K. Kanai, "Proposal for Specification of FEC-SSCS for AAL Type 5," *Contribution ATMF/95-0326 R2, ATM Forum Technical Committee*, October 1995.
- [7] Maurizio Casoni and Jonathan S. Turner, "On the Performance of Early Packet Discard," *IEEE Journal on Selected Areas in Communications*, Vol. 15, no 5, pp. 892-902, Jun. 1997.
- [8] Ion Stoica, Scott Shenker, and Hui Zhang, "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocation in High Speed Networks," in *Proceedings of SIGCOMM'98*, Sept 1998.
- [9] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, "A Survey of Active Network Research," *IEEE Communications Magazine*, pp. 80-86, January 1997.
- [10] David A. Halls and Sean G. Rooney, "Controlling the Tempest: Adaptive Management in Advanced ATM Control Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 16, N° 3, pp. 414-423, April 1998.
- [11] G. Pujolle and D. Gaïti, "ATM Flow Control Schemes Through a Multi-Agent System," *International Conference on Information Engineering '93, Proceedings of IEEE*, Volume 1, pp. 455-459, 1993.
- [12] Jiann-Liang Chen, Ying-Ping Yu, and Shinfeng Lin, "Exploiting Multi-Agent Scheme for Traffic Management in ATM Networks," *Intelligent Control (ISIC), Proceedings, IEEE*, pp. 594-599, September 1998.
- [13] Jim Hardwicke and Rob Davison, "Software Agents for ATM Performance Management," *Networks Operations and Management Symposium NOMS '98, IEEE*, Volume 2, pp. 313-321, 1998.
- [14] The ATM Forum Technical Committee. Traffic Management Specification Version 4.0 af-tm-0056.000, April 1996.
- [15] German Goldszmidt, and Yechiam Yemini, "Delegated Agents for Network Management," *IEEE Communications Magazine*, Volume 36, 3, pp. 66-70, March 1998.
- [16] F. Nait-Abdesselam, N. Agoulmine, and A. Kasiolas, "Agents Based Approach for QoS Adaptation In Distributed Multimedia Applications over ATM Networks," *IEEE International Conference on ATM, ICATM-98*, pp. 319-326, 1998.
- [17] Alex L. G. Hayzelden and John Bigham (Eds.), "Software Agents for Future Communications Systems," *Springer*, 1999.
- [18] José Luis González-Sánchez and Jordi Domingo-Pascual, "RAP: Protocol for Reliable Transfers in ATM Networks with Active Switches," *CIC'2000, Proceedings*, pp. 141-148, June 2000.
- [19] José Luis González-Sánchez and Jordi Domingo-Pascual, "TAP: Architecture for Trusted Transfers in ATM Networks with Active Switches," *IEEE ATM'2000, Proceedings*, pp. 105-112, June 2000.
- [20] J. M. Pitss, "Introduction to ATM. Design and Performance," *Ed. Wiley*, 1997.
- [21] A.L. Roginsky, L.A. Tomek and K.J. Christensen, "Analysis of ATM Cell Loss for Systems with on/off Traffic Sources," *IEE Proc. Commun.* Vol. 144. No. 3, pp. 129-134, June 1997.
- [22] Norbert Vicari and Robert Schedel, "Performance of the GFR-Service with Constant Available Bandwidth," *INFOCOM'99 Proceedings, IEEE*, pp. 567-574 vol.2, 1999.

Session 25

Mobile Communication Networks VI

Trusted and Active Protocol over a Distributed Architecture in ATM Networks with Software Agents¹

José Luis González-Sánchez (*) and Jordi Domingo-Pascual (**)

(*) University of Extremadura. Escuela Politécnica de Cáceres Avda. Universidad S/N. (10.071) Cáceres (Spain)

Phone: +34 927 257 259 Fax: +34 927 257 202 E-Mail: jlgs@unex.es

(**) Polytechnic University of Catalunya. Campus Nord, Modul D6, Jordi Girona 1-3 (08034) Barcelona (Spain)

Phone: +34 93 401 6981 Fax: +34 93 401 7055 E-Mail: jordi.domingo@ac.upc.es

Abstract-TAP (Trusted and Active PDU transfers) is a new distributed architecture and protocol for ATM networks that provides assured transfers to a set of privileged VPI/VCI. The distributed architecture proposes a new Extended AAL type 5 (EAAL-5), manages the privileged connections and offers an improvement in the performance when network connections cause some cell loss by taking advantage of the idle time in the traffic sources to carry out the retransmissions of CPCS-PDU-EAAL5. The trusted protocol is supported by our AcTMs (active ATM switch) model that we have equipped with hardware techniques and active software to achieve our objectives. Several simulations (using ON/OFF sources) demonstrate the effectiveness of the mechanism that recovers the congested PDUs locally at the congested switches with better goodput in the network. Also, the senders are alleviated of negative end-to-end retransmissions. The TAP is an active and distributed architecture in the sense that our protocol implements several active coordinated and self-collaborative software agents.

1. INTRODUCTION AND RELATED WORK

Reliability in ATM networks is provided by the Header Error Control (HEC) field of 8 bits in the header of the ATM cells and by the Cyclic Redundancy Check (CRC) in the Common Sublayer-Protocol Data Unit (CS-PDU). Error control is performed end-to-end by the terminals. The main problem is that a single cell loss causes a reassembly CRC error at AAL-5 level, which in turn leads to a retransmission of a complete PDU (i.e., IP datagram).

ATM networks experience three types of errors [1-2]: cell losses due to congestion in switches; corruption of data portions due to bit errors, and switching errors due to undetected corruption of the cell header. We note that congestion is by far the most common type of error, and here is where we want to improve the trusted transfers with TAP.

Current literature describes three basic techniques to achieve reliability: ARQ [3] (Automatic Repeat Request), FEC [4-6] (Forward Error Correction) and hybrid mechanisms of ARQ in combination with FEC.

While ARQ adds latency (due to the cost of NACK) and implosion, FEC adds overhead and thus the redundant code added by this method is useless when the network is experiencing congestion. Hence, ARQ may not be suitable for applications with requirements of low latency, and FEC performs worse in networks with low bandwidth or which experience frequent congestion. In our architecture we

adopted ARQ with NACK (using RM cells) to alleviate the effect of implosion. Support for reliable multipoint cannot be based on retransmissions from the source. In TAP, the intermediate active nodes carry out the retransmissions.

The most commonly proposed congestion control schemes to improve throughput and fairness, while minimizing delay in ATM networks, are the Random Cell Discard (RCD), Partial Packet Discard (PPD), Early Packet Discard (EPD) [7], Early Selective Packet Discard (ESPD), Fair Buffer Allocation (FBA) and Random Early Detection (RED). TAP uses a modified version of EPD (which we have denominated Early Packet Discard and Relay, EPDR) to alleviate the effect of congestion and packet fragmentation.

Nowadays, congestion control is achieved by delegated relay on end-to-end protocols, such as TCP. This is an easily implemented technique at high speeds and also simplifies the switches, but all the network is overcharged with retransmissions and does not achieve protection against egoistic traffic sources. Fair bandwidth schemes protect wellbehaved sources from misbehaved customer ones, and allow a diverse set of end-to-end congestion control mechanisms. We provide support for fair bandwidth allocations based on a delegated and modified WFQ (Weighted Fair Queueing) [8] scheme to reduce its implementation complexity.

ATM Adaptation Layer type 5 (AAL-5) has been developed to support transfers of non-assured data user frames, where the lost and corrupted Common Part Convergence Sublayer Service Data Unit (CPCS-SDU) cannot be solved with retransmission [1]. We propose EAAL-5 as an extended and enhanced native AAL-5. EAAL-5 is part of TAP that supports assured service with retransmissions and is also compatible with native AAL-5. In this paper we propose a mechanism to take advantage of the idle periods in the data sources to retransmit the Common Part Convergence Sublayer PDU (CPCS-PDU) of EAAL-5.

Active, open and programmable networks is a new technical area [9-19] to explore ways in which network elements may be dynamically reprogrammed by network managers, network operators or general users to accomplish the required QoS and other features such as customized services. This offers attractive advantages, but also important challenges in aspects such as performance, security or reliability. Hence, this is an open issue for research and development in customized routing and protocols, whether to

¹ This work is sponsored in part by the CICYT under Grant No. TEL99-1117-C03-03 and by the Regional Government of Extremadura (Education, Science and Technology Council).

move the service code (placed outside the transport network) to the network's switching nodes. Many of the advantages of active protocols are achieved by installing active nodes at strategic points. Concepts such as active networks, protocol boosters or software agents are proposed and developed for IP networks; however the proposals are insufficient for ATM networks.

We have simulated and studied transfers combined with the active switch and other non-active ATM switches to constitute a VPN (Virtual Private Network) as we can see in Fig. 1. Our goal is to use the TAP architecture to approximate the functionality of a network in which all switches do not need to implement TAP. Java Development Kit V1.2.1 has been the language and environment used to implement TAP due to the special characteristics offered by Java.

Section 2 describes TAP architecture. In sections 3 and 4, we present AcTMs, our prototype of an active switch, that support the architecture and the protocol. Section 5 describes and outlines our work currently in progress to enhance TAP. Finally we offer some concluding remarks in section 6.

II. GENERAL DESCRIPTION OF TAP ARCHITECTURE

AAL-5 was proposed [1] to reduce overhead introduced by AAL3/4. The CPCS-PDU format of native AAL-5 and EAAL-5 has equal fields. The tail of PDU has 4 fields. The CPCS-UU (User-to-User indication) field is used for the transfer of CPCS user to user information. The CPI (Common Part Indication) octet is used to align 64 bits to the CPCS-PDU tail. TAP utilizes these two octets as the PDU sequence number, which is assigned end-to-end by the EAAL-5 user. The CRC is used as in AAL-5 to detect bit errors in the CPCS-PDU. The value of CRC is calculated including all the fields of the CPCS-PDU. The sequence number in PDUs is preserved end-to-end to avoid recalculating the CRC and modifying the tail of the CPCS-PDUs.

To implement NACK we use the standard Resource Management (RM) cells, without fixed frequency but generated when a switch is congested. This is to alleviate the negative overhead effect due to a fixed number of RM cells that will waste bandwidth.

When congestion is detected, EPDR discards a PDU. Then the CCA Agent searches for the discarded PDU in the DMTE. If this PDU is not in the local DMTE, the RM Agent of the active node generates a RM cell which is transmitted backwards to the upstream active switch indicating the sequence number of the discarded PDU.

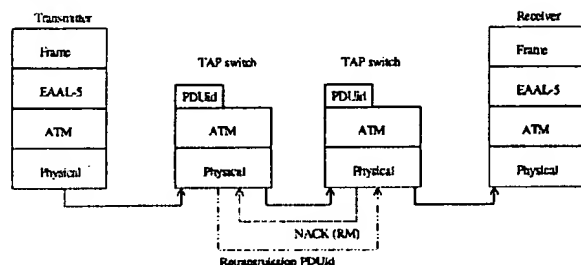


Fig. 1. Virtual Private Network with TAP.

The RM must also contain the VPI/VCI to identify the connection that has experienced discard problems. This mechanism is required to relate the sources of traffic with their I/O ports to alleviate the effect of equal values in VPI/VCI at different ports. Octets 22 to 51 of the RM cell [14] store the identifier Port/VPI/VCI/PDUid of requested PDUs.

When the RM cell arrives at an active switch, the TAP searches the requested PDU and, if it is still in DMTE, then the PDU is retransmitted, as long as the idle time for the connection is sufficient.

When a NACK (RM cell) arrives at a non-active switch this only processes the RM cell and resends it to its neighboring switch in the direction of the closest upstream active switch. The non-active switches do not have DMTE to retrieve PDUs and their function is only to send (or resend) PDUs forward to their destination and also send NACKs backwards to the active nodes.

To conclude this point we emphasize that TAP cannot offer complete reliability, but assures and recovers an important number of PDUs that otherwise would be lost by congestion. The mechanism also guarantees that there are no end-to-end retransmissions but between active peer nodes. The retransmission mechanism generates unordered PDUs at the receiver. The protocol offers two kind of service. The first one named SEQ (sequential) sorts the PDUs and when it detects a sequence failure, assumes that the PDU is lost and leaves the retrieval to protocols of upper layers (i.e. TCP). The second type of service is unordered (connectionless) and does not do any kind of sorting. These two services are offered by the proposed EAAL-5.

III. AcTMs (Active ATM Switches) NODES

Fig. 2 plots the TAP architecture including the DMTE memory and the agents. The trusted sources generate their flow that arrives at the DMTE and which is processed to the output buffer that multiplexes the cells to the next switch.

The architecture of the ATM switch is similar to an output buffered switch with VC-merging capabilities. We propose the AcTMs model of switch able to support the TAP architecture and protocol, which has the next main sections.

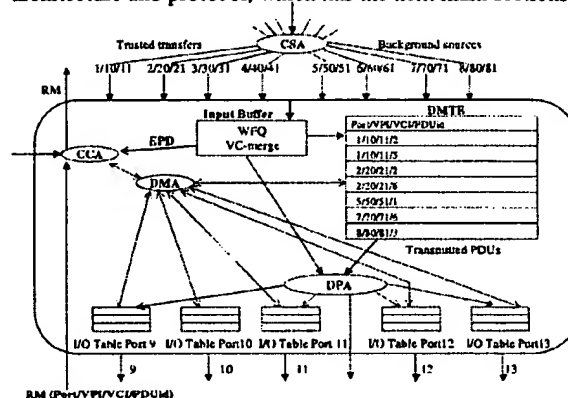


Fig. 2. TAP architecture over AcTMs switch.

A. DMTE (Dynamic Memory of Trusted EAAL-5 PDUs)

The DMTE is the key module. It behaves as a common shared memory and VC-merge buffer at the same time. The main function of this memory is to store temporarily the PDUs in the active switch after they have been transmitted to the output buffer, so that they can be requested in retransmissions. The TAP protocol keeps a copy of each PDU that arrives at the active switch (while VC merging is performed). Several PDUs are stored for each privileged connection. When a complete PDU arrives at the DMTE it is "copied" completely into the output buffer. The PDU remains in the DMTE until free space is needed for storing a new PDU.

Due to the big size of PDU-AAL-5 (up to 65,535 bytes), and the potential high number of connections (VPI/VCI), the size of the DMTE may be excessive. This is the reason why we limit the number of stored PDUs for each connection, and furthermore, only support a reduced number of privileged VCIs with trusted transfers. The traffic of a connection is more trusted when the DMTE stores more PDUs of this connection. But the size of DMTE also depends on the size of the PDUs, and we know that this is variable. We have calculated the required size of the memory to guarantee a concrete number of connections. Table I shows some of the obtained results and we can see that in order to offer trusted transfers to 1000 sources it only needs 3 Mbytes of memory that stores 2 PDUs of 1500 bytes for each privileged connection.

The TAP accesses the DMTE through an index consisting of the port number, the PDUid (which corresponds to the UU and CPI fields in AAL5) and the VPI/VCI, and we have implemented different mechanisms to optimize the management and storage of PDUs.

While a PDU from the DMTE is being retransmitted, if there is an incoming PDU with the same VPI/VCI and no free space is available in the DMTE, the incoming PDU is discarded. It is similar to a loss caused by the lack of VC-merge buffers.

B. Input buffer

As we can see in Fig. 2, the ATM cells arrive and are multiplexed over the input buffer where the cells are reassembled to build the PDUs. We use a size of 3000 octets, but the TAP simulator provides features to customize the number of cells that the input buffer stores.

TABLE I
REQUIRED DMTE SIZE TO GUARANTEE X CONNECTIONS

PDUs stored per connection	PDU size	Trusted connections	DMTE size
3	64 Kb	10	1.9 Mbytes
3	64 Kb	20	3.8 Mbytes
3	1.5 Kb	10	45 Kbytes
3	1.5 Kb	100	450 Kbytes
3	1.5 Kb	1000	4.5 Mbytes
2	1.5 Kb	1000	3 Mbytes

Every traffic management scheme requires queue management. Different methods for managing queues have different effects on traffic that flows through the queues. We know that *Work conserving systems* (FIFO) send the PDUs once the switch has completed the time of service. Hence, the server may not be idle if there are PDUs in a queue. On the other hand, the *Non-work conserving schemes* waits a random amount of time before serving the next PDU in a queue, even if PDUs are waiting in the queue. While packet switching networks use *window-based flow control* (FIFO), the high speed networks need *rate-based mechanisms* and use work-conserving services and mechanisms such as Fair Queueing (FQ). FQ waits n-1 bits times before sending and has the problem that every source has the same fraction of bandwidth. However Weighted Fair Queueing (WFQ) offers strong performance guarantees. Although algorithms designed to achieve fair bandwidth allocations provide many desirable advantages for congestion control, their implementation complexity (per-flow scheduling, per-flow buffer management and per-PDUs classification) is an important obstacle to their application in high-speed networks. We propose a WFQ scheme that works by delegation over the CSA agent as we can see in section IV.

C. Input/Output Tables

Each output port in AcTMs has its corresponding Input/Output table that stores the access index to the DMTE. When a PDU is sent to its output port, the I/O table that stores the *InPort/VPIIn/VCIIn/PDUid/VPIOut/VCIOut/OutPort* index is previously updated. This avoids equal PDUidentifiers and also provides direct access to PDUs into DMTE.

IV. MULTI-AGENT AND DISTRIBUTED SYSTEM

We bring active characteristics to TAP through hardware mechanisms and software techniques. TAP architecture includes the CCA agent (Control Congestion Agent) to manage the retransmission requests between peer active switches, the DMA agent (Dynamic Memory Agent) to recover PDU from the DMTE and also the CSA and DPA agents.

An active network is a programmable network that allows code to be loaded dynamically into network nodes at run-time. The literature on active networks studies several mechanisms to obtain advantage from active nodes. However, the proposals are insufficient for ATM networks and references [10-19] are some examples of this recent research in ATM.

There is no consensus on deciding when a network is active. There are two great tendencies: *a network is active if it incorporates active nodes with the capacity to execute a user's program*, or *if it implements mechanisms of code propagation*. The TAP architecture is active in both trends, because it provides active nodes at strategic points that implement an active protocol to allow user's code to be loaded dynamically into network nodes at run-time. TAP also provides support for code propagation in the network thanks to the RM cells. TAP is also a distributed architecture in the sense that the protocol uses several active coordinated and self-collaborative agents.

The ATM switch of our model network is an output buffered switch that just reads VPI/VCI information from arriving cells and forwards them to the corresponding output port. But we equipped this switch with active hardware (as we just see in section III) and software techniques (as now we will see). The TAP architecture uses four agents to perform the following functions.

A. CSA (Class of Service Agent)

This programmable agent supports by delegation the WFQ scheme. The TAP protocol extends the flow-based WFQ functionality to provide support for user-defined classes of service. CSA allows the definition of classes of service with a list of parameters for each class such as PCR, size of PDUs, QoS, exact amount of bandwidth, VPI/VCI identifiers, Ton, Toff, queue limit size, etc. When the list of parameters is defined in the transmitter, the CSA is coordinated with the following agents to guarantee the class of service defined end-to-end and the WFQ scheme provide congestion control and allocates bandwidth in a fair manner.

B. CCA (Control Congestion Agent)

The CCA programmable agent controls congestion based in EPDR and other algorithms that the network manager can choose. This agent monitors the output buffer and, when the occupancy is above the threshold, it discards any new incoming PDUs (packets). The last EAAL-5 cell contains the VPI and VCI in the header and the PDUid in the trailer of the EAAL-5-CPCS-PDU. The complete PDU is discarded as in EPD but the information about the VPI, VCI and PDUid is used to generate a request for the retransmission of this PDU. If the requested PDU is still in the local DMTE, it may be recovered and resent to the output buffer. Otherwise, the requests must be forwarded to the upstream active switch.

Fig. 3 shows the EPD algorithm modified to carry out the retransmissions of PDUs when a congestion is detected at input buffer (Early Packet Discard and Relay).

```

When a cell arrives at an ATM buffer:
  if the cell's VPI/VCI belongs to drop-list
    if the cell is an EOM cell
      if Queue_Length < Buffer_Size
        insert the cell into buffer
        get PDUid
        generates RM cell (Request PDUid)
      else
        discard the cell
    else
      remove the VPI/VCI from the drop-list
      discard the cell
  else
    if Queue_Length < Threshold
      insert the cell into buffer
    else if (BOM cell or (the buffer is full))
      discard the cell
      capture the VPI/VCI into drop-list
    else
      accept the cell into the single buffer
  
```

Fig. 3. Early Packet Discard and Relay Algorithm.

Another important point is that CCAs do not perform a protocol in the classical way; that is, there is only one opportunity to recover a PDU. No sliding windows or timeout retransmissions are used in this proposal.

C. DMA (Dynamic Memory Agent)

The DMA agent is the access point to DMTE memory. The CCA is coordinated with the Dynamic Memory Agent to request retransmission of the PDU to the peer active switches. The function of the CCA agent is to generate native RM cells that are transmitted backwards to the upstream active switch. Non-active switches will recognize the RM cells as TAP RM cells and will not take any action on them; they will simply forward the RM cells.

When a CCA agent receives an RM cell it takes on the task of looking for the requested PDU in the DMTE memory using *Port/VPI/VCI/PDUid* as the index. But this work is delegated over the DMA agent that uses the I/O tables to search the access index to the DMTE memory. If the PDU is still there it means that the connection cell flow presents an idle period and the PDU may be recovered. The PDU is sent to the DPA agent that dispatches it to their correspondent output port. When the PDU is not yet at DMTE the DMA agent notifies the CCA agent that is responsible for generating a RM cell backward to the previous ATM switch.

We should recall that PDUids are assigned end-to-end for each VCC and there is not any change for misinterpretation of the requested PDU. If the cell flow of the connection is dense (very short idle periods between successive PDUs) then the new incoming PDUs will use the DMTE and the "old" PDUs will be removed. We are currently working on the use of RM cells as a transport mechanism to carry out code propagation between active nodes. This code contains instructions to optimize the retransmission of PDUs in multipoint connections. The CCA agents utilize these instructions to inspect the distribution tree at width providing better goodput in retransmissions.

D. DPA (Dispatcher PDUs Agent)

This agent takes the complete PDUs of the input buffer and, when their Input/Output buffer is updated, the PDU is sent to the correct output port. DPA guarantees that the PDUs are sent completed to the output port. The congestion of the output port and also, the negative effect of merge cells belonging to different PDUs or sources, are avoided.

V. PERFORMANCE EVALUATION

Previous work [18,19] has presented and demonstrated the good behavior of RAP and TAP protocols over TAP architecture. We have simulated several software techniques to introduce active characteristics in switches. These mechanisms control and manage the privileged VCI and we will also offer an active mechanism to retrieve PDUs querying neighboring active switches and to search optimized paths when a PDU is retransmitted.

The simulation allows us to define the congestion probability in transmitters, receivers and each ATM switch. When a node is undergoing congestion, it then requests the retransmission (NACK) of the corresponding PDU.

The simulation also permits the user to introduce variable values such as ON/OFF traffic source parameters, the number of transmitters and receivers, the number of non-active switches, size of DMTE and input buffer, etc. Also, the simulator offers the operator the management of the parameters of programmable agents such as CSA and CCA.

In our simulation to analyse ATM cell loss we have used ON/OFF (bursty) traffic sources. The ON/OFF model [20,21] is used to characterise ATM traffic per unidirectional connection. Fig. 4 shows this model as a source which either actively sends (ON state) CSCP-PDU-AAL-5 data for some time T_{on} at a traffic rate R or PCR (Peak Cell Rate) or is silent (OFF state) producing no cells for some time T_{off} .

The source also periodically generates empty time slots. We use in all examples a CSR (Cell Slot Rate) of $C=353,208$ cell/s since our network model uses 155.52 Mbit/s links. When the CAR is less than the CSR, there are empty slots during the active states as we can see in Fig. 4.

The cell inter-arrival time $1/CAR$ is the unit of time for the ON state, and the mean duration in the active state is

$$T_{on} = (1 / CAR) * X_{on}. \quad (1)$$

Also, the mean duration in the silent or idle state is

$$T_{off} = (1 / CSR) * X_{off}. \quad (2)$$

Empirical studies [20] demonstrate that $T_{on} = 0.96$ seconds, and $T_{off} = 1.69$ seconds, and we use these values in the simulation, although we have used other values to analyze its effect over TAP. We use these and other formulae to implement the sources of simulations. Note that we have varied some of these parameters to analyse the behaviour of the TAP when it changes the scenario and the source traffic descriptors as we show in this section.

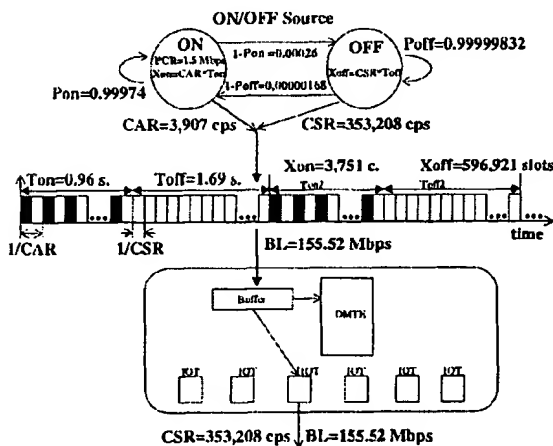


Fig. 4. Cell pattern for a single ON/OFF source and example of simulation.

Table II shows the maximum and minimum source traffic descriptors used in our simulation. We utilize a process that switches between an idle (silent) state, and the active state (sojourn time) which produces an average fixed rate of cells (between 64 Kbits/s to 25 Mbits/s) grouped in PDUs of 1,500 bytes. During the ON states this process generates cells at a cell arrival rate CAR.

Fig. 5 shows several identical sources, each operating independently over an AcTMs switch equipped with an ATM buffer of finite size X bytes and service capacity C cells/second receiving cells since the ON/OFF sources. The Peak Cell Rate is PCR cells/second, so the Mean Cell Rate (MCR) for each ON/OFF source is,

$$MCR = PCR (T_{on} / (T_{on} + T_{off})). \quad (3)$$

The probability that the source is active, or activity factor, is,

$$AF = MCR / PCR = T_{on} / (T_{on} + T_{off}). \quad (4)$$

We can calculate how many times the peak rate PCR fits into the service capacity C , denoted by S ,

$$S = C / PCR. \quad (5)$$

In this way, in the example shown in Fig. 4,

$S = 353,208 / 3,751 = 94.1$ ON/OFF trusted sources with $PCR = 1.5$ Mbps and multiplexed over a link bandwidth of 155.52 Mbps.

So, we will have enough aggregated T_{off} time to retrieve PDUs when we have not exceeded 94 multiplexed sources. This gives the maximum number of sources that we can have in the system to take advantage of the inactivity time to retransmit congested PDUs.

When TAP protocol detects that the aggregated PCR of sources exceed the service capacity C , with burst scale queuing, it does not request retransmissions of PDUs. So, this feature is accomplished by the CCA agent which, before requesting retransmissions, checks that these are possible in order to avoid waste network resources.

TABLE II
SOURCE TRAFFIC DESCRIPTORS ON/OFF

Source traffic descriptor	Parameter	Minimum	Maximum
Bandwidth Source	BS	64 kbit/s.	25 Mbit/s.
Cell arrival rate	CAR	167 cells/s.	65,105 c/s.
Cell inter-arrival time	1/CAR	6 ms.	15 μ s.
Bandwidth link	BL	155.52 Mbps.	622 Mbit/s.
Cell slot rate	CSR	353,208 cell/s.	1,412,648 cell/s.
Service time per cell	1/CSR	2.83 μ s.	0.70 μ s.
Active time period	T_{on}	0.96 s.	1 s.
Mean number of cells in an active state	X_{on}	160 cells	65,105 cells
Time in idle state	T_{off}	1.69 s.	2 s.
Mean number of empty slots in idle states	X_{off}	596,921 cell slots	2,825,296 cell slots

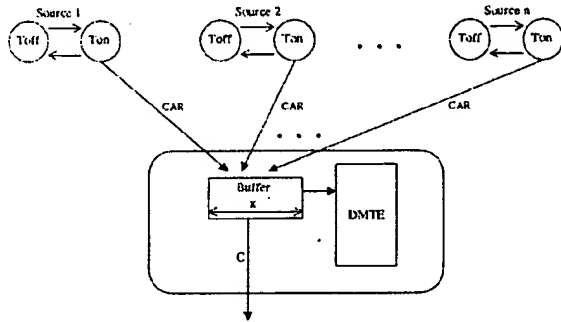


Fig. 5. Multiple ON/OFF sources.

We now report some results from the simulation of the TAP protocol.

Fig. 6 shows the effect of varying PCR between 86 and 2,667 cells per second (33,000 Kbps to 1 Mbps respectively). In this simulation we fixed the congestion probability at 10^{-3} . We use an input buffer of 3,000 octets and the DMTE stores 2 PDUs of 1,500 bytes for each connection.

The value for PCR is 64 Kbps (167 cells/s.); Ton=0.96 s.; and Toff=1.69 s, over the 50 total PDUs discarded by congestion, 50 PDUs are retrieved via TAP. Also, when PCR=56 Kbps and 33 Kbps, TAP retrieves all the congested PDUs. Thus, the performance is optimized (50 retrieved PDUs out of 50 congested PDUs) since all the lost PDUs are retrieved and there are no DMTE failures (all the requested PDUs are in the DMTE).

As we can see, when the arrival rate is low, the number of retrieved PDUs increases. When the PCR increases, 256 Kbps, TAP retrieves 48 out of 50 PDUs, but the 2 lost PDUs are not requested because the protocol detects insufficient idle time (Toff) to do the retransmission. We can see how the number of NACKs not sent (Not requested PDUs) is greater when the PCR value increases. In this way, the network is not over-charged with useless retransmissions when there is not sufficient aggregate Toff.

We note that at high PCR (1 Mbps) the number of retrieved PDUs is 47 and also the 3 not retrieved PDUs are not requested. As we can see the goodput is optimized when the number of trusted sources do not exceed the service capacity C (see Fig. 4 and 5).

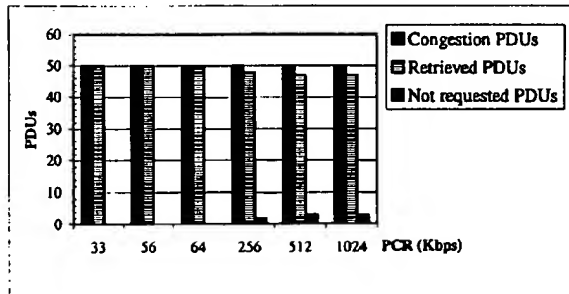


Fig. 6. Number of retrieved PDUs for different PCRs.

Fig. 7 shows the results of varying the idle time (Toff) between 0.1 and 2 seconds. We now use 10 ON/OFF sources over a bandwidth link of 25 Mbps. Each source generates 500 PDUs at PCR=1 Mbps. We fixed a Cell Loss Rate of 5 % over the total emitted PDUs and we have a constant value of 25 congested PDUs. As we can see, when the aggregated Toff is sufficient (0.5 seconds), all the congested PDUs are retrieved (25 retrieved over 25 congested). When the Toff is less than 0.5 s. the retrieved PDUs fall to 12 PDUs at 0.3 s., and 3 retrieved PDUs at 0.1 s. So we noted that, when there is insufficient Toff, the number of unretrieved PDUs increases, but TAP guarantee the goodput since the unrecoverable PDUs are not requested to avoid overcharging the network.

Another scenario consists of 1 source node, 1 active ATM switch, n non-active switches and 1 destination node. When a NACK arrives at a non-active switch, this also transfers the RM cell to the next switch. When the RM arrives at the active switch this uses the DMTE to retransmit the requested PDU. This scenario is the same as above, only the number of non-active switches varies. In this configuration we have simulated the protocol with several non-active switches and the results obtained show no changes. Only the delay in transmissions varies due to propagation times, but the index of retrieved PDUs is maintained as we have already shown.

Previous work [18,19] presents a point-multipoint configuration consisting of 1 source node, 1 active ATM switch, n non-active switches and n destination nodes. This is equal to the above basic scenario, only the number of destination nodes in multipoint connections varies. At present we are working to achieve multipoint connections to TAP. If we consider the above results we can see intuitively that the total delay will change. Also the amount of DMTE memory required increases in active switches to manage the VPI/VCI of n connections.

However, we shall now describe several aspects on which we are working to achieve better goodput.

We will consider other source traffic descriptors such as SCR (Sustainable Cell Rate) and MBS (Maximum Burst Size). With these parameters we can characterize the traffic better. Also, like most applications used in the TCP protocol for transmission data in frame based structures, we are working to implement the Guaranteed Frame Rate (GFR) [22] service class to provide a minimum service guarantee for

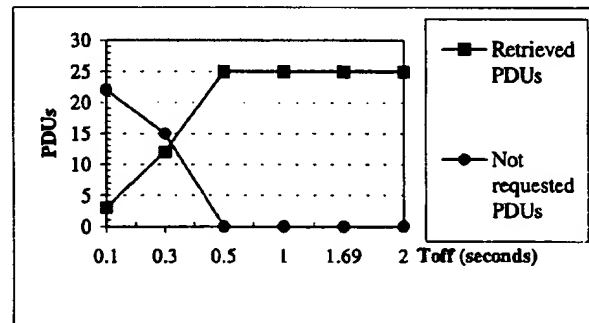


Fig. 7. Effect of variation Toff.

UBR, VBR and ABR services. In order to support GFR we will simulate sources with a Minimum Cell Rate (MCR) guarantee for a given MBS and Maximum Frame Size (MFS). With the GFR service class TAP will guarantee that is able to distinguish eligible and non-eligible frames and also to discard cells properly. We are currently working to enhance the architecture including other intelligent agents to characterize the traffic and their class of service.

VI. SUMMARY

In this paper we have presented TAP as the architecture for an active protocol that can take advantage of suitably equipped active ATM switches. TAP manages a set of privileged VCI's to improve trusted connections when the switches are congested. To achieve these active characteristics we use AcTMs (active ATM switches) equipped with little support hardware and software agents of reduced implementation complexity. We have verified that it is possible to retrieve an important number of PDUs only with DMTE and a reasonable additional complexity of the AcTMs switches supported by software agents that implement variants of EPD to solve congestions, and WFQ to achieve fair allocation. The retransmission mechanism is based on ARQ with NACK that generates RM cells to request PDUs. Our simulations demonstrate that the intuitive idea of taking advantage of silent states in ON/OFF sources is valid. Thus we can achieve better goodput and QoS in ATM networks.

REFERENCES

- [1] Recommendation I.363.5, "B-ISDN ATM Adaptation Layer Specification, Type 5," *ITU-T*, 08/1996.
- [2] Recommendation I.371.1, "Traffic Control and Congestion Control in B-ISDN," *ITU-T*, 06/1997.
- [3] Stephan Block, Ken Chen, Philippe Godlewski, and Ahmed Serhrouchni, "Design and Implementation of a Transport Layer Protocol for Reliable Multicast Communication," *Université Paris*, <http://www.enst.fr/~block/srmtp>, 1998.
- [4] Jörg Nonnenmacher, M. Lacher, M. Jung, E. Biersack, and G. Carle, "How bad is Reliable Multicast without Local Recovery?," *INFOCOM'98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE*, Vol. 3 pp. 972-979, 1998.
- [5] Ernst W. Biersack, "Performance Evaluation of Forward Error Correction in an ATM Environment," *IEEE Journal on Selected Areas in Comm.*, vol. 11, No. 4, pp. 631-640, May, 1993.
- [6] H. Esaki, G. Carle, T. Dwight, A. Guha, K. Tsunoda, and K. Kanai, "Proposal for Specification of FEC-SSCS for AAL Type 5," *Contribution ATMF/95-0326 R2, ATM Forum Technical Committee*, October 1995.
- [7] Maurizio Casoni and Jonathan S. Turner, "On the Performance of Early Packet Discard," *IEEE Journal on Selected Areas in Communications*, Vol. 15, no 5, pp. 892-902, Jun. 1997.
- [8] Ion Stoica, Scott Shenker, and Hui Zhang, "Core-Stateless Fair Queueing: Achieving Approximately Fair Bandwidth Allocation in High Speed Networks," in *Proceedings of SIGCOMM'98*, Sept 1998.
- [9] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, and Gary J. Minden, "A Survey of Active Network Research," *IEEE Communications Magazine*, pp. 80-86, January 1997.
- [10] David A. Halls and Sean G. Rooney, "Controlling the Tempest: Adaptive Management in Advanced ATM Control Architecture," *IEEE Journal on Selected Areas in Communications*, Vol. 16, N° 3, pp. 414-423, April 1998.
- [11] G. Pujolle and D. Gaïti, "ATM Flow Control Schemes Through a Multi-Agent System," *International Conference on Information Engineering'93, Proceedings of IEEE*, Volume 1, pp. 455-459, 1993.
- [12] Jiann-Liang Chen, Ying-Ping Yu, and Shinfeng Lin, "Exploiting Multi-Agent Scheme for Traffic Management in ATM Networks," *Intelligent Control (ISIC), Proceedings, IEEE*, pp. 594-599, September 1998.
- [13] Jim Hardwicke and Rob Davison, "Software Agents for ATM Performance Management," *Networks Operations and Management Symposium NOMS 98, IEEE*, Volume 2, pp. 313-321, 1998.
- [14] The ATM Forum Technical Committee. Traffic Management Specification Version 4.0 af-tm-0056.000, April 1996.
- [15] German Goldszmidt, and Yechiam Yemini, "Delegated Agents for Network Management," *IEEE Communications Magazine*, Volume 36, 3, pp. 66-70, March 1998.
- [16] F. Nait-Abdesselam, N. Agoulmine, and A. Kasiolas, "Agents Based Approach for QoS Adaptation In Distributed Multimedia Applications over ATM Networks," *IEEE International Conference on ATM, ICATM-98*, pp. 319-326, 1998.
- [17] Alex L. G. Hayzelden and John Bigham (Eds.), "Software Agents for Future Communications Systems," *Springer*, 1999.
- [18] José Luis González-Sánchez and Jordi Domingo-Pascual, "RAP: Protocol for Reliable Transfers in ATM Networks with Active Switches," *CIC'2000, Proceedings*, pp. 141-148, June 2000.
- [19] José Luis González-Sánchez and Jordi Domingo-Pascual, "TAP: Architecture for Trusted Transfers in ATM Networks with Active Switches," *IEEE ATM'2000, Proceedings*, pp. 105-112, June 2000.
- [20] J. M. Pitss, "Introduction to ATM. Design and Performance," *Ed. Wiley*, 1997.
- [21] A.L. Roginsky, L.A. Tomek and K.J. Christensen, "Analysis of ATM Cell Loss for Systems with on/off Traffic Sources," *IEE Proc. Commun.* Vol. 144. No. 3, pp. 129-134, June 1997.
- [22] Norbert Vicari and Robert Schedel, "Performance of the GFR-Service with Constant Available Bandwidth," *INFOCOM'99 Proceedings, IEEE*, pp. 567-574 vol.2, 1999.